

DETAILED ACTION

Response to Amendment

1. This office action is in response to applicant's amendment filed, 07 January 2010, of application filed, with the above serial number, on 12 June 2001 in which claims 1-21, 23, 26-29, and 31-34 have been amended, claims 22, 30, and 35-37 have been cancelled, and claims 38-39 have been added. Claims 1-21, 23-29, 31-34, and 38-39 are pending in the application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-21, 23-29, 31-34, and 38-39, are rejected under 35 U.S.C. 103(a) as being unpatentable over Albert et al (hereinafter "Albert", 6,775,692) in view of Pai et al (hereinafter "Pai", Locality-Aware Request Distribution in Cluster-based Network Servers).

Albert teaches the invention as claimed including TCP state migration and monitoring (at least Abstract).

As per Claim 1, Albert teaches in a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first server computer (forwarding agent & service manager), said first server computer

part of a plurality of server computers forming a web cluster containing information (at least col. 7, lines 36-60; col. 3, lines 22-57; forwarding agents connecting client / server clusters), said communication session established for the transfer of data contained within said information (at least col. 7, lines 36-60; col. 3, lines 22-57; forwarding agents connecting client / server clusters);

c) migrating a first TCP state of said first server computer to said selected server computer, and a second TCP state of said selected server computer to said first server computer over said control channel (at least Fig. 5; col. 14, lines 1-15; forwarding data packet).

Albert fails to explicitly teach b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, wherein the TCP handoff modules implement a TCP handoff protocol that works within kernel levels of the operating systems. However, the use and advantages for using such a system is well known to one skilled in the art at the time the invention was made as evidenced by the teachings of Pai. Pai teaches a TCP handoff protocol wherein a client establishes a connection with a front-end server, and the connection is then transparently handed to a back end server, with the handoff protocol layered on top of TCP and the subsequent connection to the back end server being forwarded by a forwarding module at the bottom of the protocol stack, for communications over HTTP persistent connections (at least Fig. 16; pp. 213 section 5-6.1). Therefore, it would have

been obvious to one of ordinary skill in the art, at the time the invention was made, to incorporate the use of Pai's system into Albert as this would further enhance Albert's system for use in load balancing and allowing the TCP connection to have very fast forwarding of acknowledgement packets and overall system performance increases.

As per Claim 2. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 1, wherein said step a) comprises the steps of:

receiving a SYN packet from said client computer at a first bottom TCP (BTCP) module located at said first server computer (at least col. 12 line 23 - col. 13 line 51);

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer (at least col. 12 line 23 - col. 13 line 51);

receiving a first SYN/ACK packet from said first TCP module (at least col. 12 line 23 - col. 13 line 51);

parsing said first TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session (at least col. 12 line 23 - col. 13 line 51; col. 19, lines 12-15);

sending said SYN/ACK packet to said client computer (at least col. 12 line 23 - col. 13 line 51);

receiving an ACK packet from said client at said first BTCP module (at least col. 12 line 23 - col. 13 line 51);

sending said ACK packet to said first TCP module (at least col. 12 line 23 - col. 13 line 51);

receiving a web request packet associated with said TCP/IP communication session at said first BTCP module at said first server computer (at least col. 12 line 23 - col. 13 line 51);

storing said SYN packet, said ACK packet and said web request packet at said first server computer (at least col. 12 line 23 - col. 13 line 51; col. 19, lines 12-15; TCP connection being established between the client, forwarding agent and server).

As per Claim 3. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 2, wherein said step b) comprises the steps of:

examining content of said web request packet (at least col. 9, lines 10-34, 45-58; service manager detailing load balancing);

determining which of said plurality of server computers can best process said web request packet, based on said content, wherein the server computer determined to be able to best process said web request packet is the selected server computer (at least col. 9, lines 10-34, 45-58; service manager detailing load balancing);

sending a handoff request packet from said first BTCP module to a second BTCP module at said selected server computer over said control channel, if said selected server computer is not said first server computer (at least col. 14 line 65 - col. 15 line 27; SYN/ACK packets);

including said SYN packet and said ACK packet in said handoff request packet (at least col. 14 line 65 - col. 15 line 27; SYN/ACK packets);

changing a destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module (at least col. 7 line 60 - col. 8 line 11; modifying addresses in header);

sending said SYN packet from said second BTCP module to a second TCP module at said selected server computer (at least col. 12 line 23 - col. 13 line 51);

receiving a second SYN/ACK packet at said second BTCP module (at least col. 12 line 23 - col. 13 line 51);

parsing said second TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session; changing a destination IP address of said ACK packet to said second IP address, at said second BTCP module (at least col. 12 line 23 - col. 13 line 51);

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session; sending said ACK packet that is updated from said second BTCP module to said second TCP module; and sending a handoff acknowledgment message from said second BTCP module to said first BTCP module (at least col. 12 line 23 - col. 13 line 51).

As per Claim 4. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 3, wherein step c) comprises:

migrating said first TCP state to said second BTCP module over said control channel by including said first initial sequence number in said handoff request packet, such that said second BTCP module can calculate said first TCP state for said first

server computer in said TCP/IP communication session (at least Fig. 5; col. 14, lines 1-15; forwarding data packet).

As per Claim 5. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 3, wherein step c) comprises:

migrating said second TCP state of said selected server computer to said first BTCP module by including said second initial sequence number in said handoff acknowledgment message, such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communication session (at least Fig. 5; col. 14, lines 1-15; forwarding data packet).

As per Claim 6. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 2, comprising the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module at the first server computer, said connection indication message sent by said first TCP module upon establishing said communication session (at least col. 15 line 36 - col. 16 line 15; col. 8, lines 17-25; http from client intercepted by service manager and forwarding agent); and

holding said connection indication message at said first UTCP module, wherein said first UTCP module and said first BTCP module provide a wrapper around said first TCP module (at least col. 15 line 36 - col. 16 line 15; col. 8, lines 17-25; Pai section 5).

As per Claim 7. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 6, wherein said method comprises the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment message to said first TCP module (at least Fig. 13; col. 12 line 23 - col. 13 line 51; col. 32, lines 46-63; TCP connection ending between the client, forwarding agent and server);

discarding said connection indication message at said first UTCP module (at least Fig. 13; col. 12 line 23 - col. 13 line 51; col. 32, lines 46-63; TCP connection ending between the client, forwarding agent and server);

receiving incoming data packets from said client at said first BTCP module (at least col. 15 line 36 - col. 16 line 15; col. 8, lines 17-25; http from client);

changing said destination addresses of said incoming data packets to said second IP address (at least col. 7 line 60 - col. 8 line 11; modifying addresses in header);

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer (at least col. 12 line 23 - col. 13 line 51; col. 19, lines 12-15); and

forwarding said updated data packets to said selected server computer (at least Fig. 5; col. 14, lines 1-15; forwarding data packet).

As per Claim 8. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 6, comprising the further steps of:

sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer (at least Fig. 13; col. 12 line 23 - col. 13 line 51; col. 32, lines 46-63; TCP connection ending between the client, forwarding agent and server);

sending incoming data packets, including said web request packet, from said client computer, received at said first BTCP module, upstream (at least Fig. 13; col. 12 line 23 - col. 13 line 51; col. 32, lines 46-63; TCP connection ending between the client, forwarding agent and server).

As per Claim 9. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 2, comprising the further steps of:

intercepting outgoing response packets from said selected server computer at a second bottom TCP (BTCP) module located at said selected server computer (at least col. 12 line 23 - col. 13 line 51);

changing, by the second BTCP module, source addresses of said response packets to a first IP address of said first server computer (at least col. 7 line 60 - col. 8 line 11; modifying addresses in header);

updating, by the second BTCP module, sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer (at least col. 12 line 23 - col. 13 line 51; col. 19, lines 12-15); and

sending said updated response packets to said client computer without passing the updated response packets through the first server computer(at least col. 7 line 60 - col. 8 line 11; modifying addresses in header; Pai figure 15 reply bypassing front-end).

As per Claim 10. Albert, along with Pai's teachings as disclosed above, teaches the method as described in Claim 1, comprising the further steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module (at least col. 32, lines 46-63; col. 8, lines 17-39; service manager monitoring packets);

understanding when said communication session is closed at said selected server computer (at least col. 32, lines 46-63; col. 8, lines 17-39; connection ends);

sending a termination message to said first server computer over said control channel (at least col. 32, lines 46-63; connection ends);

terminating said TCP/IP communication session at said first server computer by terminating a forwarding mode at said first BTCP module (at least col. 32, lines 46-63; connection ends); and

freeing data resources associated with said communication session at said first server computer (at least col. 3, lines 26-56; load balancing).

As per Claim 38. (New) The method as described in Claim 1, wherein the TCP handoff modules are dynamically loadable kernel modules, the method further comprising:

dynamically loading the TCP handoff modules in the corresponding first server computer and selected server computer without modifying the operating systems of the respective first server computer and selected server computer (Pai pg. 213, section 5 ¶ 3-4; Fig. 15; handoff taking place within the kernel, with no changes needed and server applications can run unmodified).

As per Claim 39. (New) The method as described in claim 6, wherein the first BTCP module and first UTCP module are dynamically loadable kernel modules (Pai pg. 213, section 5 ¶ 3-4; Fig. 15; handoff taking place within the kernel, with no changes needed and server applications can run unmodified).

Claims 11-21, 23-29, and 31-34, do not, in substance, add or define any additional limitations over claims 1-10 and 38-39 and therefore are rejected for similar reasons.

Response to Arguments

4. Applicant's arguments filed 07 January 2010 have been fully considered but they are not persuasive. Applicant argues Pai does not teach the handoff protocol being "dynamically loadable within TCP/IP stacks". However, the specification states that TCP handoff modules can be dynamically loadable as dynamically loadable kernel modules without service interruption and without updating the operating system, allowing for application transparency (p. 24:10-27). Pai clearly teaches this as Pai teaches the handoff protocol layered on top of TCP running on the front-end and back-end nodes and that the handoff protocol is transparent to clients and also to the server applications running on the back-end nodes (Pai pg. 213, section 5 ¶ 3-4; Fig. 15), as well as the handoff taking place within the kernel, with no changes needed and server applications can run unmodified.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Currently cited Kazemi et al, in addition to previously cited Tanenbaum et al, Aiken et al, Johnson et al, Lee et al, Wang, TCP Handoff, Brendel et al, Brendel, Vange et al, Soderberg et al, Aviani et al, and Colby et al, are cited for disclosing pertinent information related to the claimed invention. Applicants are requested to consider the prior art references for relevant teachings when responding to this office action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to GREGORY G. TODD whose telephone number is (571)272-4011. The examiner can normally be reached on Monday - Friday 9:00am-5:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571)272-4001. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/G. G. T./
Examiner, Art Unit 2457

/ARIO ETIENNE/
Supervisory Patent Examiner, Art Unit 2457